

A Near Optimal Reliable Composition Approach for Geo-Distributed Latency-Sensitive Service Chains

Dmitrii Chemodanov^{*†}, Prasad Calyam^{*}, Flavio Esposito[†]
^{*}University of Missouri-Columbia, USA; [†]Saint Louis University, USA.

Abstract—Traditionally, Network Function Virtualization uses Service Function Chaining (SFC) to *place* service functions and *chain* them with corresponding flows allocation. With the advent of Edge computing and IoT, a *reliable composition* of latency-sensitive SFCs is needed to support applications in geo-distributed cloud infrastructures. However, the optimal SFC composition in this case becomes the NP-hard integer *multi-commodity-chain* flow (MCCF) problem that has no known approximation guarantees. In this paper, we present a novel practical and near optimal SFC composition approach for geo-distributed cloud infrastructures that also admits end-to-end network QoS constraints such as latency, packet loss, etc. Specifically, we propose a novel *metapath composite variable* approach that reaches 99% optimality on average and takes seconds for practically sized integer MCCF problems of US Tier-1 (~300 nodes) and regional (~600 nodes) infrastructure providers' topologies. To ensure reliability, we compose SFCs with capacity chance-constraints and backup policies. Using trace-driven simulations comprising of challenging disaster-incident conditions, we show that our solution composes twice as many SFCs than the state-of-the-art network virtualization methods.

I. INTRODUCTION

Nowadays, Network Function Virtualization (NFV) is an attractive paradigm for network operators to dynamically *place* virtualized network functions (*e.g.*, firewalls, load-balancers, etc.), *chain* them for a service flow routing and allocate corresponding compute/network resources in a cloud infrastructure by utilizing SFCs [1]. Recently, areas such as Microservices [2], Mobile Edge Computing [3] and Computer Vision Analytics [4] have also shown benefits of adopting the SFC technology. With the advent of Edge (or Fog) computing that augments cloud Application Programming Interfaces closer to the end-user IoT devices, SFCs can be now ‘composed’ from both core and edge cloud resources forming geo-distributed chains to satisfy geo-location and latency requirements of their functions [3], [5]. An example of a geo-distributed latency-sensitive SFC which is utilized for the computer vision of a real-time object tracking pipeline is shown in Figure 1. The pre-processing and Human-Computer Interaction analysis functions are placed for a low-latency access on edge servers, and the tracking function is placed on a cloud server for the compute-intensive processing [4].

However, geo-distributed SFCs can be subject to node failures and congested network paths leading to their frequent Quality of Service (QoS) demands violations [6]. In some specific cases of natural or man-made disaster-incidents, they can be subject to severe infrastructure outages [7]. Moreover,

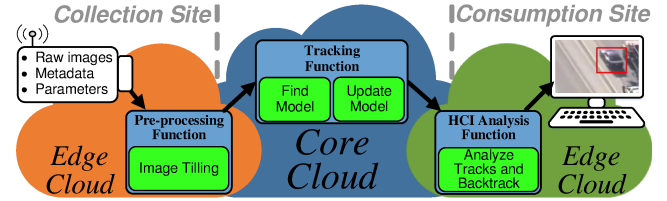


Fig. 1: Illustrative example of the geo-distributed latency-sensitive SFC used for the real-time object tracking pipeline [4].

computation and network QoS demands of SFCs can fluctuate themselves [8]. Thus, a *reliable SFC composition* is needed to cope proactively with both potential SFC demand fluctuations [8] as well as possible infrastructure outages [6], [7].

At the same time, providing a reliable service chain composition is a hard problem to solve. First of all, this is because the optimal SFC composition has known approximation guarantees only in some special cases where chaining of service functions [9] and/or their ordering [10], [11] are omitted. In the general case however, it requires solving of the NP-hard integer MCCF problem to align flow splits with supported hardware granularity [12]. It is also necessary to support cases when service functions or their associated flows are nonsplittable. This problem has no known approximation guarantees and has been previously reported as the integer NFV service distribution problem [13]. Secondly, its complexity can be further exacerbated by incorporated reliability and geo-location/latency aware mechanisms. The former aims to cope proactively with both possible infrastructure outages as well as SFC demand fluctuations, whereas the latter is needed to satisfy QoS demands of geo-distributed latency-sensitive SFCs.

Our approach: In this paper, we propose a new *reliable service chain composition* approach that can serve needs of geo-distributed latency-sensitive SFCs at high-scale. First, our approach involves ensuring reliability proactively. For this, we *compose* SFCs with capacity chance-constraints (that handle both SFC demand fluctuations as well as infrastructure outages uncertainties) and with backup policies which further complicate solution of the NP-hard integer MCCF problem. Secondly, to cope with this problem solution intractabilities, we propose a novel *metapath-based composite variable* approach that is similar to other composite variable solutions [14] in terms of its nature that aggregates multiple decisions within a single binary variable.

Contributions: In this paper, we propose a novel practical and near optimal SFC composition approach for practically sized integer MCCF problems that admit end-to-end network QoS constraints such as latency, packet loss, etc. Specifically, our contributions are the following:

[†]Corresponding and primary author (dycbt4@mail.missouri.edu).

This work has been supported by NSF awards CNS-1647084, CNS-1647182 and Coulter Foundation Translational Partnership Program. The information reported here does not reflect the views of the funding agencies.

(1) We formulate the (master) NP-hard integer MCCF problem previously adopted in NFV literature [13] but now with geo-location and latency constraints as well as with probabilistic capacity constraints for a reliable composition under uncertainty of geo-distributed latency-sensitive SFCs. (Section III)

(2) We propose our first-of-its-kind *metapath-based composite variable* approach that aggregates feasible mapping decisions of each single-link SFC segment as a set of k -constrained shortest metapaths. It then assigns SFC segments to their associated metapaths either optimally by using generalized assignment problem (GAP) [15] or suboptimally by using its (polynomial) Lagrangian relaxation counterpart. (Section IV)

(3) Using trace-driven simulations of real US Tier-1 (~ 300 nodes) and regional (~ 600 nodes) infrastructure providers' topologies, we first show how our SFC composition approach achieves 99% optimality on average. In addition, we show that it only takes time on the order of seconds for practically sized problems in contrast with the master problem solution that takes several hours. By recreating challenging disaster incident scenarios as in [7], we lastly show how our approach can compose twice as many sequentially incoming SFC requests than the state-of-the-art solutions [16], [17]. (Section V)

II. RELATED WORK

SFC is traditionally used in NFV to place a set of middleboxes and chain relevant functions to steer traffic through them [1]. Existing SFC solutions either separate the service placement from the service chaining phase [9], [10], [11], or jointly optimize both the two phases [8], [13].

SFC Optimality. In some special cases the optimal SFC is shown to have approximation guarantees [9], [10], [11]. For instance, Cohen et al. [10] and Sang et al. [11] provide near optimal approximation algorithms for the SFC problem without chaining and ordering constraints. Tomassilli et al. [9] propose the first SFC solution with the approximation guarantees which admits ordering constraints, but still omits chaining constraints. Guo et al. [18] show approximation guarantees for SFCs with both ordering and chaining constraints, but only under assumptions that available service chaining options are of polynomial size. In the general case however, when service functions need to be jointly placed and chained in a geo-distributed cloud infrastructure with a corresponding compute/network resource allocation, possible SFC compositions are of exponential size. Thus, it becomes a linear topology Virtual Network Embedding (VNE) [3], [19] and can be formulated as the (NP-hard) MCCF problem with integrality constraints with no known approximation guarantees [13]. Thus, Feng et al. [13] propose a heuristic algorithm whose preliminary evaluation results in a small-scale network settings (of ~ 10 nodes) shows promise for providing efficient solutions to the integer MCCF problem in practical settings.

In this paper, we propose the first to our knowledge practical and near optimal SFC composition approach in the general case of joint service function placement and chaining in a geo-distributed cloud infrastructure that also admits end-to-end network QoS constraints such as latency, packet loss, etc. To this aim, we propose a novel *metapath composite variable approach* which reduces a combinatorial complexity of the

(master) integer MCCF problem. As a result, our approach achieves 99% optimality on average and takes seconds to compose SFCs for practically sized problems of US Tier-1 (~ 300 nodes) and regional (~ 600 nodes) infrastructure providers' topologies, where master problem solution takes hours using a High Performance Computing cloud server.

SFC Reliability. With the advent of edge networking and growing number of latency sensitive services, recent works also consider problems of geo-distributed [20] and edge SFC [5]. Although these works mainly focuses on the new load balancing and latency optimization techniques, they omit an important reliability aspect of geo-distributed latency-sensitive SFCs. The closest works related to ours is [8] and [17]. Fei et al. [8] propose a prediction-based approach that proactively handles SFC demand fluctuations. However, their approach does not account for network/infrastructure outages that mainly cause service function failures [6]. At the same time, Spinnewyn et al. [17] propose a SFC solution that ensures a sufficient infrastructure reliability, but neither proactively nor reactively handles SFC demand fluctuations.

In contrast to [8] and [17], our reliable composition scheme uniquely ensures reliability of geo-distributed latency-sensitive SFCs via use of chance-constraints and backup policies to cope with both SFC demand fluctuations and infrastructure outages.

III. MODELING RELIABLE SERVICE CHAIN COMPOSITION

In this section, we define the problem of joint *SFC composition* that can be formulated as the integer MCCF problem for an augmented cloud infrastructure graph [13] which is a generalization of a well-known multi-commodity flow problem [19]. To proactively ensure reliability of a SFC composition, we use backup policies as well as probabilistic 'chance' capacity constraints instead of deterministic ones. Thus, we use a chance-constrained programming [21]. We also extend this problem with geo-location and latency constraints to satisfy all QoS demands of geo-distributed latency-sensitive SFCs.

Objective and example of the online chain composition. Based on providers' policies, the service chain composition problem can be used to minimize (expected) values of different fitness functions $F_{\mathbb{E}}$. One example of common fitness functions is an additive function of service chain demands and corresponding physical resource capacity ratios. Such function is known to best balance the physical network load [19]. In most cases service chain requests can be unknown in advance, and using the load balancing fitness function allows to increase the acceptance ratio of these requests. Such optimization is also known as the 'online optimization' [16], [19].

Figure 2 shows an example of the online SFC composition that minimizes the network load balancing function: by minimizing a sum of SFC demands and corresponding physical resource capacity ratios, for $a-b-c$ service chain we achieve its minimum value $F_{\mathbb{E}} = \frac{8}{10} + \frac{2.1}{5} + \frac{8}{10} + \frac{1}{5} + \frac{4}{5} = 3$. As a result, we compose this service chain request with X , Y and A physical nodes (*e.g.*, servers) to place a , b and c services, respectively. To enable service communications $a-b$ and $b-c$, we chain them with $X-B-Y$ and $Y-B$ physical paths, respectively. This also allows us to compose the subsequent $d-e$ request.

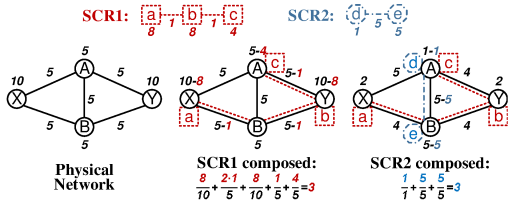


Fig. 2: Illustrative example of the *online composition* of service chain requests (SCR) on top of the capacitated physical network (numbers indicate service demands and corresponding resource capacities).

In the rest of this paper, our objective is to minimize the expected value of the load balancing fitness function $F_{\mathbb{E}}$ formally defined below (see Equation 1) for the case of SFC demand and available physical resource uncertainties.

Service chain composition sets and variables. We model each SFC $a \in A$ as a chain graph $G^a = (N_V^a, E_V^a)$. A service G^a is composed by a set N_V^a of services and a set E_V^a of corresponding service communications (or service links) representing logical network connectivities among elements in N_V^a . Moreover, each SFC a has a set of backup resources B^a . We then model the physical infrastructure on which the service functions run as a physical network graph $G = (N_S, E_S)$, composed by a set N_S of substrate nodes and a set E_S of substrate edges.

We define two types of binary variables: one for the service chain link mapping, and another for (node) service mapping. Particularly, let binary variable $f_{ij}^{st}(b, a) = 1$ if a flow for $st \in E_V^a$ service link of a backup $b \in B^a$ of a SFC $a \in A$ is assigned to the physical edge $ij \in E_S$, i.e., $f_{ij}^{st}(b, a) = 1$, or 0 otherwise. Furthermore, let binary variable $x_i^s(b, a) = 1$ if a service $s \in N_V^a$ of a backup $b \in B^a$ of a SFC $a \in A$ is assigned to the physical node $i \in N_S$, i.e., $x_i^s(b, a) = 1$, and 0 otherwise. Having sets and variables defined, we now can formulate the online service chain composition problem under uncertainty using integer MCCF problem.

Problem 1 (online SFC composition under uncertainty). *Given a set of SFCs represented as graphs $G^a = (N_V^a, E_V^a)$ and a physical network graph $G = (N_S, E_S)$, the online service chain composition problem under SFC demands and available physical resources uncertainties can be formulated as follows:*

$$\min F_{\mathbb{E}} = \sum_{a \in A} \sum_{b \in B^a} \left(\sum_{s \in N_V^a} \sum_{\tau \in T} \sum_{i \in N_S} \mathbb{E} \left[\frac{D_{s\tau}^{a\tau}}{C_i^\tau} \right] x_i^s(b, a) + \sum_{st \in E_V^a} \sum_{ij \in E_S} \mathbb{E} \left[\frac{D_{st}^a}{C_{ij}} \right] f_{ij}^{st}(b, a) \right) \quad (1)$$

subject to

Service Placement Constraints:

$$\sum_{i \in N_S} x_i^s(b, a) = 1, \forall s \in N_V^a, b \in B^a, a \in A \quad (2)$$

$$\mathbb{P} \left[\sum_{a \in A} \sum_{b \in B^a} \sum_{s \in N_V^a} D_{s\tau}^{a\tau} x_i^s(b, a) \leq C_i^\tau \right] \geq R, \forall i \in N_S, \tau \in T \quad (3)$$

Service Chaining Constraints:

$$\sum_{j \in N_S} f_{ij}^{st}(b, a) - \sum_{j \in N_S} f_{ji}^{st}(b, a) = x_i^s(b, a) - x_i^t(b, a), \quad (4)$$

$$\forall i \in N_S, st \in E_V^a, b \in B^a, a \in A$$

TABLE I: Symbols and Notations

Service Chain Composition: Sets	
A	\triangleq Set of SFCs that needs to be composed
B^a	\triangleq Set of backups for the SFC a
N_V^a	\triangleq Set of service functions composing the SFC a
E_V^a	\triangleq Set of service communications in the SFC a that create a service chain
\mathcal{P}_a^{st}	\triangleq Set of metapaths for the st (single-link) service chain segment of SFC a
T	\triangleq Set of QoS demands for service functions such as CPU, memory, storage, etc
K	\triangleq Set of end-to-end network QoS demands for SFC communications such as latency, losses, jitter, etc
N_S	\triangleq Set of physical nodes within the infrastructure
E_S	\triangleq Set of physical links within the infrastructure
Service Chain Composition: Variables	
$x_i^s(b, a)$	\triangleq Binary variable that equals to 1 if the service s backup b for the SFC a is placed on the physical node i
$f_{ij}^{st}(b, a)$	\triangleq Binary variable that equals to 1 if the communication backup b between services s and t for the SFC a is placed on the physical link ij
$f_{ijk}^{st}(b, a)$	\triangleq Binary variable that equals to 1 if the single-link chain segment st is assigned to the metapath P_{ijk}^{st} of the SFC a backup b
$D_s^{a\tau}$	\triangleq Random variable that corresponds to the SFC a service s SLO demand $\tau \in T$
D_{st}^a	\triangleq Random variable that corresponds to the communication bandwidth demand between services s and t for the SFC a
C_i^τ	\triangleq Discrete random variable that corresponds to the possible physical node i capacity of $\tau \in T$ resource, i.e., $C_i^\tau \in \{0, C_i^\tau\}$
C_{ij}	\triangleq Discrete random variable that corresponds to the possible physical link ij capacity, i.e., $C_{ij} \in \{0, C_{ij}\}$
Service Chain Composition: Parameters	
gd_{si}^a	\triangleq Parameter that corresponds to the geographical distance between the desired location of the service s in the SFC a and the physical node i location
\mathcal{GD}_s^a	\triangleq Parameter that corresponds to the maximum allowable geographical distance between the desired location of the service s for the SFC a and some physical node location
w_{ij}^k	\triangleq Parameter that corresponds to the additive weight of the physical link ij (or multiplicative when composed with \log function) of the service communication end-to-end $k \in K$ QoS constraint (e.g., latency)
\mathcal{K}_{st}^{ak}	\triangleq Parameter that corresponds to the communication end-to-end $k \in K$ QoS constraint (e.g., latency) between services s and t of the SFC a
R	\triangleq SFC reliability probability that a chance-constraint will be satisfied
K_α	\triangleq Constant in a standard Normal distribution table corresponding to desired α probability
$\mu_{s\tau}^{a\tau}$	\triangleq Expected (or mean) value of the SFC a service s QoS demand $\tau \in T$
μ_{st}^a	\triangleq Expected (or mean) value of the the communication bandwidth demand between services s and t for the SFC a
$\sigma_{s\tau}^{a\tau}$	\triangleq Variance of the SFC a service s SLO demand $\tau \in T$
σ_{st}^a	\triangleq Variance of the the communication bandwidth demand between services s and t for the SFC a
C_i^τ	\triangleq Physical node i capacity of $\tau \in T$ resource (e.g., CPU)
C_{ij}	\triangleq Physical link ij capacity

$$\mathbb{P} \left[\sum_{a \in A} \sum_{b \in B^a} \sum_{st \in E_V^a} D_{st}^a f_{ij}^{st}(b, a) \leq C_{ij} \right] \geq R, \forall ij \in E_S \quad (5)$$

Specific QoS Constraints (Geo-Location, Latency, etc.):

$$gd_{si}^a x_i^s(b, a) \leq \mathcal{GD}_s^a, \forall i \in N_S, s \in N_V^a, b \in B^a, a \in A, \quad (6)$$

$$\sum_{ij \in E_S} w_{ij}^k f_{ij}^{st}(b, a) \leq \mathcal{K}_{st}^{ak}, \forall st \in E_V^a, b \in B^a, a \in A, k \in K \quad (7)$$

Additional Policy Constraints (e.g., No-Consolidation):

$$\sum_{b \in B^a} \sum_{s \in N_V^a} x_i^s(b, a) \leq 1, \forall i \in N_S, a \in A \quad (8)$$

where symbols and notations of sets, parameters, variables and functions are summarized in Table I.

SFC composition constraints discussion. Minimization of $F_{\mathbb{E}}$ in Equation 1 is subject to a set of constraints which contains both — basic composition constraints and constraints specific to the geo-distributed latency-sensitive SFCs. The basic constraints include service *placement* for a specified

number of duplicates (Equation 2), service *chaining* or well-known multi-commodity flow constraints (Equation 4). Additional policy constraints for service chain composition problem are also acceptable. One such example is a common ‘no consolidated service placement’ constraint that prohibits placement of two or more different services (or their backups) belonging to the same service chain onto one physical node (Equation 8). Note that this policy further complicates a combinatorial complexity of the (NP-hard) integer MCF problem. In contrast to prior SFC composition problems [13], [17], we now use probabilistic physical node and link capacity constraints to ensure that physical resources satisfy SFC QoS demands given some acceptable risk (Equations 3 and 5).

The specific geo-distributed latency-sensitive SFC constraints include physical node geo-location and service communication end-to-end network QoS constraints such as latency, packet loss, etc. (Equation 7).

Objective and chance-constraint deterministic equivalents. Let us consider physical node capacity constraints (see Equation 16). The risk of a physical node outage is a random discrete variable, hence the probability of a physical node capacity feasibility is:

$$\mathbb{P} \left[\sum_{a \in Ab \in B^a} \sum_{s \in N_V^a} \mathbf{D}_s^{a\tau} x_i^s(b, a) \leq C_i^\tau \right] \cong \mathbb{P} \left[\sum_{a \in Ab \in B^a} \sum_{s \in N_V^a} \mathbf{D}_s^{a\tau} x_i^s(b, a) \leq C_i^\tau \right] \cdot P_i,$$

where $\bar{P}_i = 1 - P_i$ is the outage risk of a physical node i . Thus, our physical node capacity chance-constraint is the following:

$$\mathbb{P} \left[\sum_{a \in Ab \in B^a} \sum_{s \in N_V^a} \mathbf{D}_s^{a\tau} x_i^s(b, a) \leq C_i^\tau \right] \geq \frac{R}{P_i} \quad (9)$$

We assume that SFC demands follow a normal distribution $\mathbf{D}_s^{a\tau} \sim \mathcal{N}(\mu_s^{a\tau}, \sigma_s^{a\tau^2})$. Hence, $\sum_{a \in As \in N_V^a} \mathcal{N}(\mu_s^{a\tau}, \sigma_s^{a\tau^2}) = \mathcal{N}(\sum_{a \in As \in N_V^a} \mu_s^{a\tau}, \sum_{a \in As \in N_V^a} (\sigma_s^{a\tau^2} + \sum_{a_1 \neq a \in As_1 \neq s \in N_V^a} \rho_{ss_1}^{aa_1\tau} \sigma_s^{a\tau} \sigma_{s_1}^{a_1\tau}))$. Moreover, we assume the worst-case scenario that all SFC demands are strongly correlated (*i.e.*, $\rho_{ss_1}^{aa_1\tau} = 1$).¹ Thus, given any two SFCs a and a_1 , their total deviation is $\sigma = \sqrt{\sigma_a^2 + \sigma_{a_1}^2 + 2\sigma_a\sigma_{a_1}} = \sqrt{(\sigma_a + \sigma_{a_1})^2} = \sigma_a + \sigma_{a_1}$.

As a results, we can substitute our objective and chance-constraints in Equations 3 and 5 with the following linear deterministic equivalents:

$$F = \sum_{a \in Ab \in B^a} \left(\sum_{s \in N_V^a} \sum_{\tau \in T \in N_S} \frac{\mu_s^{a\tau} + K_{\frac{R}{P_i}} \sigma_s^{a\tau}}{C_i^\tau} x_i^s(b, a) + \sum_{st \in E_V^a} \sum_{ij \in E_S} \frac{\mu_{st}^a + K_{\frac{R}{P_{ij}}} \sigma_{st}^a}{C_{ij}} f_{ij}^{st}(b, a) \right) \quad (10)$$

¹Note that such assumption is also valid when SFC demands are not correlated or their correlation coefficients are unknown. In these cases, linear deterministic equivalents of chance-constraints always satisfy availability requirements R and can result in even higher (actual) availability at expense of worse resource utilization.

$$\sum_{a \in Ab \in B^a} \sum_{s \in N_V^a} \left(\mu_s^{a\tau} + K_{\frac{R}{P_i}} \sigma_s^{a\tau} \right) x_i^s(a, b) \leq \begin{cases} C_i^\tau, & R < P_i \\ 0, & \text{otherwise} \end{cases}, \quad (11)$$

$$\forall i \in N_S, \tau \in T$$

$$\sum_{a \in Ab \in B^a} \sum_{st \in E_V^a} \left(\mu_{st}^a + K_{\frac{R}{P_{ij}}} \sigma_{st}^a \right) f_{ij}^{st}(a, b) \leq \begin{cases} C_{ij}, & R < P_{ij} \\ 0, & \text{otherwise} \end{cases}, \quad (12)$$

$$\forall ij \in E_S$$

where $K_{\frac{R}{P_{ij}}}$ is a constant in a standard Normal distribution table, $R \in (0, 1)$ is the service chain reliability that reflects an acceptable risk, and P_i (or P_{ij}) is a probability that physical node i (or link ij) is available, *i.e.*, $C_i^\tau = C_i^\tau$ (or $C_{ij} = C_{ij}^\tau$). Note that if SFC demands do not follow normal distribution, deterministic equivalents of Problem 1 objective and chance-constraints can be different.

SFC reliability and chance-constraints discussion. It is known that chance-constrained programming is less effective than multi-stage recourse programming to model uncertainties [21]. This is because to provide the same reliability level chance-constrained SFC composition under-provision more physical resources than its recourse programming alternative. On the other hand, solving a recourse program for the SFC composition is intractable even with moderately small network sizes. This is due to the fact that solving it requires computations over an exponential number of scenarios, *i.e.*, the problem is equivalent to an integer program of exponential size [17]. To avoid considering an exponential number of scenarios, we use a policy-based reliability for the SFC composition instead. Specifically, we allow for policy specifications of chance-constraints acceptable risks and service backups.

For instance, by decreasing an acceptable risk and/or increasing number of backups, we can leverage the overall probability of a SFC disruption that requires its re-composition (*e.g.*, migration of virtual resources) during its maintenance. For example, given a risk of 5%, *i.e.*, $R = 0.95$, and 5 services for a single SFC, the lower bound probability that its demands will be satisfied is $P_{lb} = R^5 = 0.95^5 \approx 0.77$ not considering inter-service communication demands and not allowing backup resources. Thus, approximately in 1 out of 5 cases the service chain needs to be re-composed. Alternatively, if we at least duplicate the service chain physical resources (*i.e.*, compose 2 service chain backups), the lower bound probability that SFC demands will be satisfied by at least one of the duplicates becomes $P_{lb} = 1 - (1 - R^5)^2 \approx 0.95$. As a result, the service chain needs to be re-composed only in 1 out of 20 cases. However, the tighter reliability policies (*i.e.*, the lower acceptable risk or the higher number of backups), the less feasible solutions are available as well as the worse objective value of the optimal solution, and thus, the worse performance of the online service chain composition. We show such reliability/performance trade-offs of our approach using trace-driven simulations in Section V.

MCCF-based SFC composition intractabilities. When deterministic equivalents of the objective in Equation 1 as well as of capacity chance-constraints are known, we can use any integer programming solver (*e.g.*, CPLEX [22]) for Problem 1 to reliably compose all (known at a time) service chain requests. However, due to NP-hardness of this composition, the solution

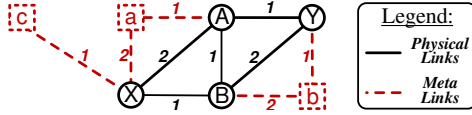


Fig. 3: Illustrative example of the augmented with metalinks physical network which represent feasible service a , b and c placements; numbers indicate fitness function values - red and black values annotate service placement and service chaining via some physical link, respectively.

can be intractable for large-scale cloud/edge infrastructures. To improve its scalability limitations, existing column generation [16], heuristic [17] and metaheuristic [17] approaches can be used (often at expense of the master problem optimality). In the next section, we propose a near optimal *metapath composite variable* approach that simplifies a combinatorial complexity of the SFC composition outlined in Problem 1.

IV. SERVICE CHAIN COMPOSITION VIA METAPATHS

In this section, we aim to simplify the combinatorial complexity of the integer MCF-based SFC composition problem. Similarly to existing composite variable schemes [14], our goal is to create a binary variable that composes multiple (preferably close to optimal) decisions. To this end, we build upon a known result in optimization theory: all network flow problems can be decomposed into paths and cycles [23]. We first introduce our notion of *metapath* and its relevance to the constrained shortest path problem [24], [25], [26]. We then use the constrained shortest metapaths to create variables with composite decisions for the SFC composition problem and discuss scalability improvements of this approach.

Metalinks and metapaths. Before defining the metapath, it is useful to introduce the idea of ‘metalinks’. Metalinks have been widely adopted in prior NFV/VNE literature to solve optimally graph matching problems [13], [16]. A metalink is an augmentation link in a network graph. In our case, it represents the (potential) *feasible* placement of some service a on some physical node A , as shown in Figure 3. Formally, we have:

Definition 1 (metalink). A link si for SFC $a \in A$ belongs to the set of metalinks E_M^a if and only if the service $s \in N_V^a$ of SFC a can be placed onto the node $i \in N_S$.

Building on the definition of a metalink, we can define a metapath as the path that connects any two services through the physical network augmented with metalinks. For example, consider following metapaths $a - A - Y - b$ and $a - A - B - b$ shown in Figure 3. Formally, we have:

Definition 2 (metapath). The path P_{ij}^{st} is a metapath between services s and t for SFC $a \in A$ if and only if $\forall kl \in P_{ij}^{st} : kl \in E_S \vee kl \in \{si, tj\}$.

Intuitively, metapath P_{ij}^{st} is formed by exactly two metalinks that connect s and t to the physical network and an arbitrary number of physical links $kl \in E_S$.

Constrained shortest metapaths. Having defined metapaths, let us consider a simple case of the SFC composition problem - composition of a single-link chain (*i.e.*, two services connected via a single virtual link): the optimal composition of a single-link chain can be seen as the *constrained shortest (meta)path* problem that connects two services via the augmented physical network, where all physical links have arbitrary

fitness values of a service chaining (virtual link mapping) and all metalinks have arbitrary fitness values of a service placement divided by the number of neighboring services (*i.e.*, by 1 for a single-link chain). In our example, shown in Figure 3, the optimal single-link SFC $a - b$ composition can be represented by the constrained shortest metapath $a - A - Y - b$ that satisfies all SFC composition constraints with the overall fitness function of 3. Further, we prove our intuition formally: **Theorem IV.1.** (The optimal single-link SFC composition) *The optimal single-link chain composition is the constrained shortest metapath.*

Proof. Assume the contrary. Let $L_1(s, t)$ be the optimal objective value of the single-link service chain st composition and $L_2(s, t)$ be a length of the constrained shortest metapath P_2 . We need to show that $L_1 \neq L_2$:

Case 1 ($L_1 < L_2$): In this case, $L_1(s, t)$ solution is mapping of services s and t to physical nodes i and j , respectively, and a service link st to a physical path $P(i, j)$ as defined in the service chain composition problem. Without loss of generality, we can assume that the optimal solution of the service chain composition problem is feasible. Hence, s and t mappings are si and tj metalinks by Definition 1, respectively. Furthermore, let us define the path $P_1 = P(si, P(i, j), tj)$ which by Definition 2 is a metapath. As the optimal solution is feasible, P_1 satisfies all constraints of the single-link chain st composition. Hence, P_1 is a constrained metapath whose length $L_1(s, t)$ is shorter than $L_2(s, t)$ contradicting that P_2 is the constrained shortest metapath.

Case 2 ($L_1 > L_2$): In this case, we can present metapath P_2 as $P_2 = P(si, P(i, j), tj)$, where si and tj are metalinks, and $P(i, j)$ is a physical path (see Definition 1). Let us map services s and t on physical nodes i and j , respectively, and service link st on a physical path $P(i, j)$. As P_2 is the constrained metapath, this mapping is feasible with the objective value $L_2(s, t)$ less than $L_1(s, t)$ contradicting that $L_1(s, t)$ is the optimal objective value of the single-link service chain st composition. ■

Corollary IV.1. (The optimal single-link SFC composition complexity) *The optimal single-link SFC composition has a pseudo-polynomial complexity.*

Proof. Based on Theorem IV.1, the optimal single-link SFC is the constrained shortest metapath which is by Definition 2 the constrained shortest path in the augmented network graph. However, it is known that the constrained-shortest path can be found in pseudo-polynomial time [26]. ■

We conclude that constrained shortest metapaths are good candidates to perform composite decisions, *i.e.*, to optimally decide on a single-link SFC composition in terms of its services placement and chaining with a single binary variable.

Multiple-link chain composition via metapath. While observing Figure 3, we can notice how using only a single constrained shortest metapath per a single-link segment of a multiple-link SFC $a - b - c$ can lead to an unfeasible composition: as the optimal $a - b$ composition is $a - A - Y - b$ metapath, and the optimal $b - c$ composition is $b - B - X - c$ metapath - service b has to be simultaneously placed on Y and B physical nodes. Thus, we cannot *stitch* these metapath,

and we need to find more than one constrained shortest metapath per a single-link chain. In our composite variable approach, we find k -constrained shortest metapaths (to create k binary variables) per each single-link segment of a multi-link service chain. To find metapaths any constrained shortest path algorithm can be used [24], [25], [26]. In this paper, we build upon the path finder proposed in our prior work that is an order of magnitude faster than recent solutions [24].

To further benefit from constrained shortest metapaths and simplify the chain composition problem, we offload its constraints (either fully or partially) to either metalinks or the path finder. Specifically, *geo-location and an arbitrary number of end-to-end network (e.g., latency) QoS constraints can be fully offloaded to metalinks and to the path finder, respectively*. At the same time, capacity constraints of the SFC composition problem are global and can be only partially offloaded. Once k -constrained shortest paths have been found for each single-link service chain segment, we can solve GAP problem [15] to assign each single-link chain segment to exactly one constrained shortest metapath and *stitch* these metapaths as described below.

Allowable fitness functions for metapath-based variables.

In general, fitness functions qualify for our *metapath composite variable* approach if they are comprised from either additive or multiplicative terms. The above requirement fits for most SFC objectives [1], and other objectives can also qualify if well-behaved (e.g., if their single-link chain fitness values can be minimized by a path finder). As the load balancing fitness function $F_{\mathbb{E}}$ in Equation 1 qualifies, we compute its single-link chain value $\mathbb{E}[F_{ijk}^{sta}]$ for k metapath as following:

$$\begin{aligned} \mathbb{E}[F_{ijk}^{sta}] &= \sum_{\tau \in T} \mathbb{E}\left[\frac{\mathbf{D}_s^{a\tau}}{\mathbf{C}_i^\tau}\right] / \deg(s) + \\ &+ \sum_{st \in E_V^a} \sum_{\substack{uv \in E_S: \\ uv \in P_{ijk}^{sta}}} \mathbb{E}\left[\frac{\mathbf{D}_{st}^a}{\mathbf{C}_{uv}^\tau}\right] + \sum_{\tau \in T} \mathbb{E}\left[\frac{\mathbf{D}_t^{a\tau}}{\mathbf{C}_j^\tau}\right] / \deg(t), \end{aligned} \quad (13)$$

where $\deg(s)$ corresponds to the service s degree, i.e., $\deg(s) = 1$ for $s \in \{in, out\}$ services that handle input and processed output data of SFCs, respectively; and $\deg(s) = 2$ otherwise. Note that in NFV *in* and *out* are dummy services that corresponds to the flow source and sink physical nodes and have no computation demands. The first and the last terms represent the fitness values of metalinks, and the middle term corresponds to the sum of physical links' fitness values.

Problem 2 (SFC composition via metapaths). *Given a set of SFCs $a \in A$ represented as graphs $G^a = (N_V^a, E_V^a)$, a physical network graph $G = (N_S, E_S)$, and having set of k -constrained shortest metapaths $P_{ijk}^{sta} \in \mathcal{P}_a^{st}$ and their corresponding fitness function values F_{ijk}^{sta} found for each virtual link $st \in E_V^a$ in the SFC a , let a binary variable $f_{ijk}^{st}(b, a) = 1$ if the single-link chain segment st is assigned to the metapath P_{ijk}^{sta} of the backup $b \in B^a$ of SFC $a \in A$, or 0 otherwise. The SFC composition problem via metapaths can be formulated as follows:*

$$\min \sum_{a \in A} \sum_{b \in B^a} \sum_{st \in E_V^a} \sum_{P_{ijk}^{sta} \in \mathcal{P}_a^{st}} \mathbb{E}[F_{ijk}^{sta}] f_{ijk}^{st}(b, a) \quad (14)$$

subject to

Metapath Stitching (Assignment) Constraints:

$$\sum_{P_{ijk}^{sta} \in \mathcal{P}_a^{st}} f_{ijk}^{st}(b, a) - \sum_{P_{jik}^{tsa} \in \mathcal{P}_a^{ts}} f_{jik}^{ts}(b, a) = \begin{cases} -1, & t = in \\ 1, & t = out \\ 0, & otherwise \end{cases} \quad (15)$$

$$\forall t \in \{in, out\} \vee tj \in E_M^a, b \in B^a, a \in A$$

Node Capacity Chance-Constraints:

$$\begin{aligned} \mathbb{P} \left[\sum_{a \in A} \sum_{b \in B^a} \sum_{t \in N_V^a} \mathbf{D}_t^{a\tau} / \deg(t) \left(\sum_{P_{ijk}^{sta} \in \mathcal{P}_a^{st}} f_{ijk}^{st}(b, a) + \right. \right. \\ \left. \left. + \sum_{P_{jik}^{tsa} \in \mathcal{P}_a^{ts}} f_{jik}^{ts}(b, a) \right) \leq \mathbf{C}_j^\tau \right] \geq R, \forall j \in N_S, \tau \in T \end{aligned} \quad (16)$$

Link Capacity Chance-Constraints:

$$\mathbb{P} \left[\sum_{a \in A} \sum_{b \in B^a} \sum_{st \in E_V^a} \sum_{\substack{P_{ijk}^{sta} \in \mathcal{P}_a^{st}: \\ uv \in P_{ijk}^{sta}}} \mathbf{D}_{st}^a f_{ijk}^{st}(b, a) \leq \mathbf{C}_{uv} \right] \geq R, \quad (17)$$

$$\forall uv \in E_S$$

where symbols and notations of sets, parameters, variables and functions are summarized in Table I.

We remark that *in* and *out* are services that handles input and processed output data of SFCs, respectively. Note also that deterministic equivalents for the objective coefficients and capacity constraints in Equations 14, 16 and 17 are similar to deterministic equivalents of Problem 1.

A. SFC Composition via Lagrangian Relaxation

Aside from solving the NP-hard (GAP) Problem 2, we also propose its better scalable alternative. In particular, we solve the GAP using its polynomial Lagrangian relaxation by compromising both its optimality and feasibility guarantees [15]. *Our approach:* Problem 2 has two types of constraints - stitching (assignment) and capacity constraints. The assignment constraints (Equation 15) represent flow conservation constraints for metalinks $tj \in E_M^a$. Hence, these constraints form the totally unimodular constraint matrix. When having the linear objective function (Equation 14), this property allows us to relax integrality constraints on $f_{ijk}^{st}(b, a)$ variable in the incapacitated service chain composition case (when capacity constraints are omitted). As a result, we can solve the above problem using (polynomial) Linear Programming. **Lower Bound Algorithm.** Similarly to [27], we use the unimodularity property benefits and push capacity constraints (see Equations 11 and 12) to the objective. To this end, let us denote $g_1^{\tau j} = R - \mathbb{P}_j^\tau$ and $g_2^{uv} = R - \mathbb{P}_{uv}$ functions for each constraint in Equations 16 and 17, respectively. Let us define $u_1^{\tau j}$ and u_2^{uv} as the Lagrangian multipliers specified for each iteration of the subgradient method [27]; we now can define (deterministic) Lagrangian weights as following:

$$\begin{aligned} w_{ijk}^{sta} &= F_{ijk}^{sta} + u_1^{\tau j} \left((\mu_s^{a\tau} + K_{\frac{R}{P_i}} \sigma_s^{a\tau}) / \deg(s) + \right. \\ &\left. + (\mu_t^{a\tau} + K_{\frac{R}{P_j}} \sigma_t^{a\tau}) / \deg(t) \right) + \sum_{uv \in P_{ijk}^{sta}} u_2^{uv} \left(\mu_{st}^a + K_{\frac{R}{P_{uv}}} \sigma_{st}^a \right) \end{aligned} \quad (18)$$

We then can solve the following linear program \mathcal{L} with any LP solver:

$$\mathcal{L} = \min \left(\sum_{a \in A} \sum_{b \in B^a} \sum_{st \in E_V^a} \sum_{P_{ijk}^{sta} \in \mathcal{P}_a^{sta}} w_{ijk}^{sta} f_{ijk}^{sta}(b, a) + \right. \\ \left. - \sum_{j \in N_S} \sum_{\tau \in T} u_1^{\tau j} \cdot \begin{cases} C_j^{\tau}, R \leq P_j \\ 0, R > P_j \end{cases} - \sum_{uv \in E_s} u_2^{uv} \cdot \begin{cases} C_{uv}, R \leq P_{uv} \\ 0, R > P_{uv} \end{cases} \right) \quad (19)$$

subject to constraints in Equation 15. Note that to improve LB while solving \mathcal{L} , we can also fix all variables $f_{ijk}^{sta} = 0$ whose node (or link) mappings do not satisfy reliability, i.e., if $R > P_i$ or $R > P_j$ (or if $\exists uv \in P_{ijk}^{sta} : R > P_{uv}$).

If solution of \mathcal{L} satisfies GAP capacity constraints, we can stop and report optimal (or suboptimal) solution to GAP. However, if \mathcal{L} solution is unfeasible to the primal GAP problem, we can project it back to the feasible space using some polynomial heuristic algorithm to get an upper bound (UB) of the primal GAP problem.

Upper Bound Algorithm. In this paper, we propose a new (polynomial) greedy regret lower bound replication (GRLBR) algorithm that we found fast enough for our large scale GAP problem with flow assignment constraints. We build our GRLBR algorithm upon both lower bound replication and greedy regret algorithms proposed earlier in [27], and its pseudo code is outlined in Algorithm 1.

GRLBR starts by detecting the largest regret service chain segment st of SFC a' (lines 5-12), i.e., the segment with the largest difference between the first best and the second best corresponding lagrangian weights w_{ijk}^{sta} for its potential feasible assignments. If there are no feasible metapaths assignments for st of a that satisfy both assignment and capacity constraints (see Equations 15, 16 and 17), we stop and report no feasible solution (lines 6-8). Once, st of a' is found, we add it to the priority queue $Q_{a'}$ based on its langrangian weight $w_{ijk}^{sta'}$ (line 13). We then retrieve and remove the head of this queue and try to map it to the LB metapath solution first (lines 19-20), or to the lowest lagrangian weight metapath $P_{ijk'}^{sta'}$ (lines 22-23), or report no feasible solution and terminate, otherwise (lines 17-18). Finally, we allocate corresponding metapath solution resources for the service chain st segment of a' and add all its adjacent segments (lines 25-26). Once $Q_{a'}$ is empty, all service chain segments of SFC a' for its backup b' have been placed. We then mark b' backup of SFC a' as mapped and remove it from further consideration by GRLBR (lines 28-31). Note that at any time $Q_{a'}$ contains only two elements due to a linear service chain topology.

Subgradient method. Having LB and UB algorithms outlined, we use them within the general subgradient method to iteratively improve LB and UB as in [27]. To this end we start with zero u_1 and u_2 lagrangian multiplier vectors. At each iteration we track if LB solution is feasible, and if so we terminate our subgradient algorithm. Moreover, if LB has been improved, i.e., if $LB_{new} > LB$, and LB is not feasible, we project LB solution back to the feasible space with our GRLBR algorithm to obtain new UB_{new} solution and update existing UB solution if $UB_{new} < UB$. If $\frac{\|UB - LB\|}{\|LB\|} < \epsilon$ or number of iterations is exceeded, we terminate the subgradient

Algorithm 1: GRLBR

Input: $\hat{f}_{ijk}^{sta}(a, b)$:= solution of \mathcal{L}_2 ; w_{ijk}^{sta} := lagrangian weights; $P_{ijk}^{sta} \in \mathcal{P}_a^{sta}$:= set of k-constrained shortest metapaths and their corresponding fitness values F_{ijk}^{sta} found for each virtual link $st \in E_V^a$

Output: UB := upper bound to GAP problem; $f_{ijk}^{sta}(a, b)$:= feasible solution to GAP problem

```

1 begin
  /* Step 0: initialize */
2    $A' \leftarrow A$ 
3    $B'^a \leftarrow B^a, \forall a \in A$ 
4   while  $A' \neq \emptyset$  do
    /* Step 1: find highest regret virtual link  $sta'$  */
5     forall  $st \in E^a$  and  $a \in A'$  do
6       if  $\nexists P_{ijk}^{sta} : P_{ijk}^{sta}$  is feasible then
7         terminate and report no feasible solution
8       end
9        $ijk'_{sta} \leftarrow \arg \min \{w_{ijk}^{sta} : P_{ijk}^{sta} \text{ is feasible}\}$ 
10       $\rho_{sta} \leftarrow \min \{w_{ijk}^{sta} - w_{ijk'}^{sta}(sta) : P_{ijk}^{sta} \text{ is feasible, } ijk \neq ijk'_{sta}\}$ 
11    end
12     $sta' \leftarrow \arg \max_{st \in E^a, a \in A'} \{\rho_{sta}\}$ 
  /* Step 2: allocate all service chain segments that contains  $sta'$  */
13  Put  $sta'$  to the priority queue  $Q_{a'} \leftarrow \{sta', w_{ijk}^{sta'}\}$ 
14   $b' \leftarrow \min \{B'^{a'}\}$ 
15  while  $Q_{a'} \neq \emptyset$  do
16     $\hat{st} \leftarrow$  retrieve and remove  $Q_{a'}$ 's head
17    if  $\nexists P_{ijk}^{sta'} : P_{ijk}^{sta'}$  is feasible then
18      terminate and report no feasible solution
19    else if  $P_{ijk}^{sta'} : \hat{f}_{ijk}^{sta'}(a', b') == 1$  is feasible then
20       $UB \leftarrow UB + F_{ijk}^{sta'}$ 
21    else
22       $P_{ijk'}^{sta'} \leftarrow \arg \min \{w_{ijk}^{sta'} : P_{ijk}^{sta'} \text{ is feasible}\}$ 
23       $UB \leftarrow UB + F_{ijk'}^{sta'}$ 
24    end
25    allocate corresponding physical resources for  $\hat{st}$ 
26    add adjacent virtual links of  $\hat{st}$  and their best lagrangian weights to  $Q_{a'}$ 
27  end
  /* Step 3: mark  $b'$  backup of  $a'$  SFC as allocated and go to Step 1 */
28   $B'^{a'} \leftarrow B'^{a'} - b'$ 
29  if  $B'^{a'} \in \emptyset$  then
30     $A' \leftarrow A' - a'$ 
31  end
32 end
33 end

```

algorithm. At the end of each iteration u_1 and u_2 are calculated w.r.t. to their objective gradient. More implementation details as well as best practices on the subgradient method can be found in [27].

V. PERFORMANCE EVALUATION

In this section, we evaluate performance of our reliable SFC composition approach under challenging disaster incident conditions that can cause severe infrastructure outages [7]. Thus, we evaluate its performance against the state-of-the-art NFV/VNE solutions of the (master) integrer MCCF problem.

General Settings. For our simulations, we use an *HPC Cloud server* with two *Intel Xeon E5-2683 v3* 14-core CPUs at 2.00 GHz (total 56 virtual cores), 256GB RAM, and running the Ubuntu 16.04 allocated in NSF CloudLab platform [30]. We solve math programs with IBM ILOG CPLEX [22]. We use both Internet Topology Zoo [28] and Atlas [29] databases to re-create the US Tier1 and regional providers' networks as shown in Figure 4. We assume that each topology has nodes and links with uniformly distributed computation capacity

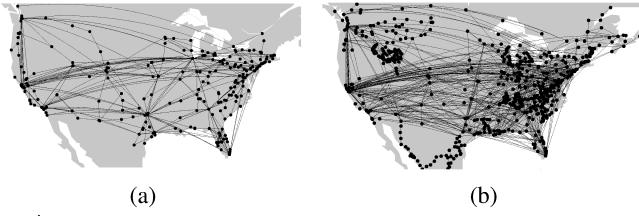


Fig. 4: Simulation data sets: (a) network infrastructure that spans 7 Tier-1 US providers and comprises of 286 Point of Presence (PoP) nodes and 534 links; and (b) network infrastructure that spans 56 regional US providers and comprises of 596 PoP nodes and 1253 links.

from 5 to 50 TFlops and bandwidth from 1 to 10 Gbps, respectively. Note that the lowerbound 5 TFlops performance can simulate limited network edge servers, whereas 50 TFlops can simulate HPC cloud servers. Moreover, we assume that latency of each physical link is proportional to its propagation delay computed as its geographical length divided by the speed of light in fiber. Finally, we compute physical resources' outage risk w.r.t. to the geographical proximity to the disaster incident epicenter as discussed in [7]. All our results show 95% confidence intervals, and our randomness lays both in SFC requests and in disaster incident events.

SFC request settings. We generate a pool of 50 SFCs composed by 2 to 20 services, unless stated differently. Based on a common object tracking application [4], each SFC has equal chances to express its either High-Performance (HPC) or regular computing demands shown in Table II. As in [7], we assume a strong correlation between SFC demands and the disaster incident intensity. Using natural disaster data sets and their associated infrastructure outage risks specified in [7], we use the following *geo-location policies*: All services handling incoming raw data must be placed within a range of two disaster incident region radiuses from its epicenter. When there are no disaster incidents, these services as well as services that output processing data have to be placed within 200 miles out of the random geographic locations picked within the US.

TABLE II: An example set of demands for different SFC types

Demands Type	Expected Computation Demand per Function	Expected Data Collection Size	Expected Comp. Time	Expected Data Rate
Regular	0.5-5 TFlops	1-10 GB	10-100 ms/frame	10-100 Mbps
HPC	1-10 TFlops	10-100 GB	10-100 ms/frame	0.1 - 1 Gbps

Composition Metrics. We compare performance of our metapath-based SFC composition in Problem 2 (referred as *MpSC*) against its (polynomial) Lagrangian relaxation counterpart (referred as *MpLG*). We also compare *MpSC* against the VNE/NFV state-of-the-art solutions of the (master) integer MCCF problem (*i.e.*, Problem 1): IBM CPLEX branch-and-bound version [22] (optimal, but has the highest combinatorial complexity), branch-and-price column generation [16] and recent isomorphism detection [17] approaches (suboptimal, but have lower combinatorial complexities). We refer to the branch-and-bound solution of the master problem as *Opt*, to the column (or path in case of SFCs) generation approach as *PgSC*, and to the isomorphism detection as *IsoSC*.

Assumptions. To evaluate the *online* optimization performance of our approach, we assume the most difficult case: all SFC requests arrive sequentially (*i.e.*, unknown in advance) and do not allow a service consolidation, *i.e.*, only one service in a chain can be placed onto the same physical server.

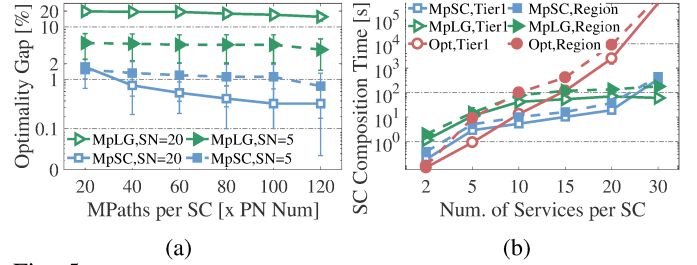


Fig. 5: Service chain (SC) composition optimality gap (a) and time (b) results in presence of no disaster incidents ($R = 0$).

Results. Our evaluation results fall under three salient thrusts of findings: (i) *our metapath approach gains more than 99% optimality on average and is up to 3 orders of magnitude faster than the master problem solution*; (ii) *our metapath approach can secure up to 2 times more SFCs in comparison to the state-of-the-art NFV/VNE approaches under challenging disaster incident conditions*; and (iii) *policies allow to trade-off between a SFC reliability and its composition optimality*.

We assess the performance of our SFC composition algorithms by specifying the fraction of successfully composed SFCs over the total requested chains (*composition ratio*). Similarly, we assess the reliability of these algorithms by computing a fraction of the number of composed SFCs disrupted during disaster incidents over the total number of composed SFCs (*disruption ratio*). In addition, we also use an *optimality gap* metric which we define as a gap in % between *Opt* and all other algorithms. Finally, we use a *composition time* metric to access *scalability* performance of our metapath approach.

MpSC gains more than 99% optimality on average and is up to 3 orders of magnitude faster than Opt. To estimate the baseline performance of our metapath approach, we first assume a scenario without disasters (*i.e.*, no outage risks and $R = 0$) where capacity chance constraints becomes deterministic yielding the largest feasible space (*i.e.*, leading to a higher combinatorial complexity of the master problem). Figure 5a shows how its optimality depends on the number of generated metapaths per single service chain (SC) request. We can see how, when this number exceeds ≥ 80 times the number of physical nodes, the performance of *MpSC* flattens. On the other hand, when either the SFC size increases or the reliability requirements get tighter in a disaster incident scene, *MpSC* achieves optimality most of the time and shows 99% optimality on average. Note also how *MpLG* shows significantly worse performance with respect to *MpSC*; this is due to use of greedy heuristics used to recover feasible solutions. However, *MpLG* can be beneficial for large SFCs (of ≥ 25 services) as it is polynomial.

For the rest of our evaluation, we fix the number of metapaths generated per SFC request to 80 and 120 times the number of physical nodes for *MpSC* and *MpLG*, respectively. These values of metapaths are picked to allow both *MpSC* and *MpLG* to compose large SFCs. For instance, with these settings *MpSC* is almost three orders of magnitude faster than the optimal solution (*Opt*) as shown in Figure 5b. For small SFCs (*i.e.*, ≤ 5 services) we found, however, no significant scalability improvements of *MpSC* and *MpLG* over *Opt*. This is due to the fact that generating metapaths is time-consuming expensive. Thus, for small service chains,

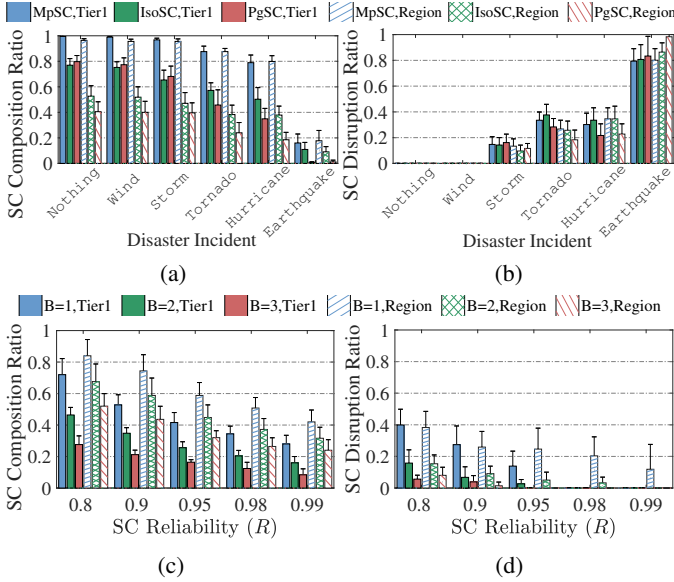


Fig. 6: Service chain (SC) composition ratio (a,c) and disruption ratio (b,c) results under different natural disaster-incidents with reliability $R = 0.8$ (first row), and $MpSC$ results under hurricane disaster-incidents (second row).

it is recommended to avoid use of metapath-based composite variables and merely consider the *Opt* policy instead. For the rest of our evaluation, we only use the *MpSC* service chain composition algorithm.

***MpSC* can secure up to 2 times more SFCs than PathGen and IsoSC under challenging disaster-incident conditions.** Furthermore, we can see how our *MpSC* outperforms *PgSC* and *IsoSC* by securing up to 2 times more SFCs under challenging disaster-incident conditions of tornadoes and hurricanes as shown in Figures 6a and 6b with the service chain reliability $R = 0.8$. This is due to the fact that *MpSC* reaches the optimality most of the time while being sufficiently scalable. At the same time, *PgSC* is limited by the performance of the SFC composition algorithm (that commonly uses a two-stage composition) to get the initial feasible solution [16]. Moreover, it is also known that column generation approaches such as *PgSC* converge slowly to the optimal for integer problems [15]. In contrast to *PgSC*, *IsoSC* doesn't need an initial feasible solution, but can fail to find one or not converge to the optimal solution for the predefined amount of iterations [17].

Policy-based SFC reliability trade-offs. Further, to achieve a desired level of reliability during SFC composition (*i.e.*, proactively), the capacity chance-constraints acceptable risk (*i.e.*, $1 - R$) and/or the number of backups policies can be adjusted appropriately. As shown for *MpSC* in Figures 6c and 6d, increasing either chance-constraints reliability R or the number of backups decreases the number of composed SFCs by either prohibiting more physical resources for allocation or utilizing more physical resources for SFC backups. On the other hand, such a strategy can significantly minimize the number of disrupted SFCs, therefore minimizing their outages.

VI. CONCLUSION

In this paper, we presented the *reliable SFC composition* approach for geo-distributed latency-sensitive SFCs. To the best of our knowledge, we present the first practical and near

optimal approach for the general NP-hard SFC composition case [13]. To ensure reliability of SFCs, we handle both their demand fluctuations and possible infrastructure outages during the composition via use of capacity chance-constraints and service backups policies. We have addressed NP-hardness limitations of the (master) integer MCCF-based SFC composition problem by proposing a novel *metapath composite variable approach* that uses either (NP-hard) GAP or its (polynomial) Lagrangian relaxation counterpart. Using realistic trace-driven simulations with US Tier-1 and regional infrastructure topologies, we have shown that our metapath composite variable approach reaches 99% optimality on average, is up to 3 orders of magnitude faster than the master problem solution for practically sized problems and can compose twice as many SFCs than related NFV/VNE methods.

REFERENCES

- [1] D. Bhamare, et al., "A survey on service function chaining", *Elsevier JNCA*, 2016.
- [2] Y. Niu, et al., "Load balancing across microservices", *Proc. of IEEE INFOCOM*, 2018.
- [3] R. Yu, et al., "Application provisioning in fog computing-enabled IoTs: a network perspective", *Proc. of IEEE INFOCOM*, 2018.
- [4] R. Gargees, et al., "Incident-Supporting Visual Cloud Computing Utilizing Software-Defined Networking", *IEEE TCSVT*, 2017.
- [5] R. Cziva, et al., "Dynamic, latency-optimal VNF placement at the network edge", *Proc. of IEEE INFOCOM*, 2018.
- [6] R. Potharaju et al., "Demystifying the dark side of the middle: a field study of middlebox failures in datacenters", *Proc. of ACM IMC*, 2013.
- [7] B. Eriksson, et al., "Riskroute: a framework for mitigating network outage threats", *Proc. of ACM CoNEXT*, 2013.
- [8] X. Fei, et al., "Adaptive VNF scaling and flow routing with proactive demand prediction", *Proc. of IEEE INFOCOM*, 2018.
- [9] A. Tomassilli, et al., "Provably efficient algorithms for placement of SFCs with ordering constraints", *Proc. of IEEE INFOCOM*, 2018.
- [10] R. Cohen, et al., "Near optimal placement of virtual network functions", *Proc. of IEEE INFOCOM*, 2015.
- [11] Y. Sang, et al., "Provably efficient algorithms for joint placement and allocation of virtual network functions", *Proc. of IEEE INFOCOM*, 2017.
- [12] S. Jain, et al., "B4: Experience with a globally-deployed software defined WAN", *Proc. of ACM SIGCOMM CCR*, 2013.
- [13] H. Feng, et al., "Approximation algorithms for the NFV service distribution problem", *Proc. of IEEE INFOCOM*, 2017.
- [14] C. Barnhart, "Airline fleet assignment with enhanced revenue modeling", *INFORMS Operations Research*, 2009.
- [15] M. L. Fisher, "The Lagrangian relaxation method for solving integer programming problems", *Management science*, 2004.
- [16] R. Mijumbi, et al., "A path generation approach to embedding of virtual networks", *IEEE TNSM*, 2015.
- [17] B. Spinnewyn, et al., "Resilient application placement for geo-distributed cloud networks", *Elsevier JNCA*, 2017.
- [18] L. Guo, et al., "Joint placement and routing of network function chains in data centers", *Proc. of IEEE INFOCOM*, 2018.
- [19] M. Chowdhury, et al., "Virtual network embedding with coordinated node and link mapping", *Proc. of INFOCOM*, 2009.
- [20] X. Fei, et al., "Towards load-balanced VNF assignment in geo-distributed NFV infrastructure", *Proc. of IEEE/ACM IWQoS*, 2017.
- [21] J.R. Birge, F. Louveaux, "Introduction to stochastic programming", *Springer Science & Business Media*, 2011.
- [22] IBM CPLEX solver - <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html>; Last accessed Jul. 2018.
- [23] R. K. Ahuja, "Network flows: theory, algorithms, and applications", *Pearson Education*, 2017.
- [24] D. Chemodanov, et al., "A Constrained Shortest Path Scheme for Virtual Network Service Management", *IEEE TNSM*, 2018.
- [25] X. Chen, et al., "Multi-criteria Routing in Networks with Path Choices", *Proc. of IEEE ICNP*, 2015.
- [26] P. Van Mieghem, F. Kuipers, "Concepts of exact QoS routing algorithms", *IEEE/ACM TON*, 2004.
- [27] V. Jeet, et al., "Lagrangian relaxation guided problem space search heuristics for generalized assignment problems", *Elsevier EJOR*, 2007.
- [28] S. Knight, et al., "The Internet Topology Zoo", *IEEE JSAC*, 2011.
- [29] R. Durairajan, et al., "Internet atlas: a geographic database of the internet", *Proc. of ACM HotPlanet*, 2013.
- [30] R. Ricci, E. Eide, "Introducing CloudLab: Scientific infrastructure for advancing cloud architectures and applications", *USENIX Login*, 2014.